REMARKS/ARGUMENTS

This Amendment is filed in response to the final Office Action dated July 25, 2007. Applicants have canceled Claims 1-8, 10-12, and 16-18, and amended various claims. Following the amendments, Claims 9, 13-15, and 19-23 remain pending in the application.

Examiner Interview

Applicants appreciate the Examiner's time in participating in a telephone interview with Applicant's representative on October 23, 2007. In the interview, proposed amendments to independent Claims 9, 15, and 19 were discussed in light of the cited references in the final Office Action dated July 25, 2007. During the discussion, the Examiner agreed that the proposed amendments to independent Claims 9 and 15 distinguish the claims from the cited references. In addition, the Examiner requested that Applicants further amend Claim 19 in order to clarify the language reciting a data field for optionally storing a control-specific identify flag. Further amendments to Claim 19 are included in this Amendment.

Claim Rejection – 35 U.S.C. § 103

In the Office Action:

(1) Claims 9 and 12-14 were rejected as obvious under 35 U.S.C. § 103(a) in light of U.S. Patent No. 5,475,843 to Halviatti et al. ("*Halviatti*") in view of U.S. Patent No. 5,634,098 to Janniro et al. ("*Janniro*") and in further view of U.S. Published Patent Application No. 2003/0052917 to Dubovsky ("*Dubovsky*");

(2) Claims 15 and 17 were rejected as obvious under 35 U.S.C. § 103(a) in light of *Halviatti* in view of *Janniro* and in further view of *Dubovsky*; and

(3) Claims 19-23 were rejected as obvious under 35 U.S.C. § 103(a) in light of *Halviatti* in view of *Janniro* and in further view of U.S. Patent No. 5,600,789 to Parker et al. ("*Parker*").

Each of the above are addressed below.

Independent Claim 9

On Page 2 of the Office Action, independent Claim 9 has been rejected as obvious under 35 U.S.C. § 103(a) in light of U.S. Patent No. 5,475,843 to Halviatti et al. ("*Halviatti*") in view of U.S. Patent No. 5,634,098 to Janniro et al. ("*Janniro*") and in further view of U.S. Published Patent Application No. 2003/0052917 to Dubovsky ("*Dubovsky*"). Although Applicants do not necessarily agree with the current rejection of Claim 9, in order to facilitate prosecution of the present application, Applicants have amended Claim 9 to further distinguish the claimed invention from the cited prior art. For example, Applicants have amended Claim 9 to specify a system for testing an application running on a target device comprising a software test tool "***wherein, in response to a user changing from a first value set to a second value set during execution of said test script, the software test tool is further configured to automatically delete all configuration item values associated with said first value set and reload said configuration items with corresponding values associated with said second value set; and continue executing said test script using said configuration item values associated with said second value set***."

Applicants note that *Halviatti* discloses a system that provides automated testing of Graphical User Interface (GUI) applications. Specifically, the system includes: (1) a Script Engine; (2) Application Translation Units (ATUs); (3) a Message Engine; (4) a Model Manager; and (5) Application-specific Testing Models (ATMs). (See Col. 21, lines 35-55 and Col. 22, lines 23-50). Each ATM is a high-level model for a specific component of the application being tested, such as File Open dialog, and describes the actual component it represents in terms of Generic Element Models (GEM). (See Col. 22, lines 5-8). A GEM encapsulates the behavior of interface objects such as push buttons, checkboxes, list-boxes, etc. (See Col. 22, lines 9-11). The ATUs are responsible for processing events received from the GUI application and translating the events into messages. (See Col. 9, lines 1-36). For example, translating the response related to performing the File Open dialog and producing a corresponding message. The Message Engine then receives the message and forwards the message, if appropriate, to the Model Manger, which manages the testing. (See Col. 9, line 45 to Col. 10, line, and Col. 22, line 23-50).

On Page 3 of the Office Action, the Examiner concedes that *Halviatti* does not specifically disclose a configuration table having a plurality of user-defined configurations, each said configuration comprising a collection of value sets corresponding to respective configuration item groups, each said value set being one of a plurality of possible value sets that are selectable in association with said respective item group, each said value set comprising a collection of one or more related configuration items and corresponding values for said related items. However, the Examiner asserts that *Janniro* does disclose such a feature.

Applicants note that *Janniro* describes a method and apparatus for performing automated testing software wherein a plurality of directories are connected to form a hierarchical directory structure that includes files associated with software programs to be tested. (See Abstract). These files include: (1) source code files; (2) test input files; (3) expected output files; (4) environment files; and (5) environment configuration files. (See Col. 5, lines 58-60). A specific test is performed by invoking a test engine and passing the name of an environment file and the name of a test to the test engine. (Col. 9, lines 2-10, Steps 402 and 404 in FIG. 4). The test engine then modifies the environment based on the environment file and further modifies the environment based on any applicable environment configuration files. (Col. 9, lines 24-27 and Col. 10, lines 7-15, Steps 406 and 412 in FIG. 4). The application of environment configuration files is test-specific and is based on the location of such files in the hierarchical directory structure. (Col. 9, lines 39-46 and Col. 10, lines 51-57). Once the test engine has modified the environment, the engine will then run the specified test based on the current values of the environment variables. (See Col. 10, lines 32-36, Step 418 in FIG. 4).

Thus, Applicants note that *Janniro* does provide environment configuration files that can be utilized to change the configuration environment of a GUI application prior to running a particular test (i.e. executing a particular test file) however the configuration environment parameters are ***static*** once the testing process has begun. On the other hand, the configuration manager of Claim 9 allows the user to change from a first value set to a second value set ***during*** the execution of a test script, which in turn, automatically modifies the configuration items during testing. This flexibility is not provided in *Janniro*.

Accordingly, in order to provide better clarity of this distinction, Claim 9 has been amended to include the feature "***wherein, in response to a user changing from a first value set to a second value set during execution of said test script, the software test tool is further configured to automatically delete all configuration item values associated with said first value set and reload said configuration items with corresponding values associated with said second value set; and continue executing said test script using said configuration item values associated with said second value set***."

In addition, *Dubovsky* describes methods and apparatus for testing computer programs. (See [0002]). In particular, the methods and apparatus of *Dubovsky* include: (1) a scriptable GUI test tool that generates a GUI map; (2) an environment definition file; (3) a test data file; and (4) an automated test engine. (See [0044]). The GUI map generated by the test tool includes the logical names and physical attributes for each GUI object of a GUI application (examples of GUI objects are check boxes, radio buttons, and text fields). (See [0045] and [0046]). A separate environment definition file is provided for each feature of the GUI. (See [0046]). The test data file provides a number of tests and the file is organized into rows of data where each row defines a single test and each column represents a parameter. (See [0020] and [0047]). In addition, a separate test engine script is provided for each GUI application to be tested. (See [0048]). The test engine runs the test engine script, which interacts with the test tool and the GUI map, and in addition, the test engine is driven by the test data file and calls upon the GUI map and environment definition file during execution. (See *Id*. and FIG. 1).

Though *Dubovsky* discloses an environment definition file that is called upon during testing wherein the file is used to define each feature of the GUI, the file is saved as a delimited text file and is managed via a spreadsheet. (See [0019] and [0046]). Thus, once testing has started the file is static and cannot be changed. In contrast, the configuration manager of Claim 9 allows the user to change from a first value set to a second value set ***during*** the execution of a test script, which in turn, automatically modifies the configuration items during testing. This flexibility is not provided in *Dubovsky*.

Thus, Applicants respectfully assert that *Halviatti, Janniro,* and *Dubovsky* fails to teach or suggest a system for testing an application running on a target device comprising a software

test tool "***wherein, in response to a user changing from a first value set to a second value set during execution of said test script, the software test tool is further configured to automatically delete all configuration item values associated with said first value set and reload said configuration items with corresponding values associated with said second value set; and continue executing said test script using said configuration item values associated with said second value set***," as recited by independent Claim 9. In fact, Applicants respectfully point out that *Halviatti*, *Janniro*, and *Dubovsky* exemplify the problem Claim 9 overcomes. Accordingly, Applicants respectfully request the Examiner withdraw the rejection of independent Claim 9.

Independent Claim 15

On Page 6 of the Office Action, independent Claim 15 has been rejected as obvious under 35 U.S.C. § 103(a) in light of U.S. Patent No. 5,475,843 to Halviatti et al. ("*Halviatti*") in view of U.S. Patent No. 5,634,098 to Janniro et al. ("*Janniro*") and in further view of U.S. Published Patent Application No. 2003/0052917 to Dubovsky ("*Dubovsky*"). Although Applicants do not necessarily agree with the current rejection of Claim 15, in order to facilitate prosecution of the present application, Applicants have amended Claim 15 to further distinguish the claimed invention from the cited prior art. For example, Applicants have amended Claim 15 to specify a method of testing an application that is operable to execute in multiple languages and platform configurations "***wherein, in response to changing from a first value set to a second value set during execution of said test script: automatically deleting all configuration item values associated with said first value set; automatically reloading said configuration items with corresponding values associated with said second value set; and continue executing said test script using said configuration item values associated with said second value set***."

On Page 8 of the Office Action, the Examiner concedes that *Halviatti* does not specifically disclose a configuration table having a plurality of user-defined configurations, each said configuration comprising a collection of value sets corresponding to respective configuration item groups, each said value set being one of a plurality of possible value sets that are selectable in association with said respective item group, each said value set comprising a

collection of one or more related configuration items and corresponding values for said related items. However, the Examiner asserts that *Janniro* does disclose such a feature.

As previously discussed, Applicants note that *Janniro* does provide environment configuration files that can be utilized to change the configuration environment of a GUI application prior to running a particular test (i.e. executing a particular test file) however the configuration environment parameters are ***static*** once the testing process has begun. On the other hand, the method of Claim 15 automatically deletes all configuration item values associated with a first value set and automatically reloads the configuration items with corresponding values associated with a second value in response to changing from the first value set to the second value set ***during*** the execution of a test script. As previously shown, this flexibility is not provided in *Janniro*.

Accordingly, in order to provide better clarity of this distinction, Claim 15 has been amended to include the feature "***wherein, in response to changing from a first value set to a second value set during execution of said test script: automatically deleting all configuration item values associated with said first value set; automatically reloading said configuration items with corresponding values associated with said second value set; and continue executing said test script using said configuration item values associated with said second value set***."

In addition, as previously discussed, *Dubovsky* discloses an environment definition file that is called upon during testing wherein the file is used to define each feature of the GUI; the file is saved as a delimited text file and is managed via a spreadsheet. (See [0019] and [0046]). Thus, once testing has started the file is static and cannot be changed. In contrast, the method of Claim 15 automatically deletes all configuration item values associated with a first value set and automatically reloads the configuration items with corresponding values associated with a second value in response to changing from the first value set to the second value set ***during*** the execution of a test script. This flexibility is not provided in *Dubovsky*.

Thus, Applicants respectfully assert that *Halviatti, Janniro,* and *Dubovsky* fails to teach or suggest a method of testing an application that is operable to execute in multiple languages and platform configurations "***wherein, in response to changing from a first value set to a second value set during execution of said test script: automatically deleting all configuration***

item values associated with said first value set; automatically reloading said configuration items with corresponding values associated with said second value set; and continue executing said test script using said configuration item values associated with said second value set," as recited by independent Claim 15. In fact, Applicants respectfully point out that *Halviatti*, *Janniro*, and *Dubovsky* exemplify the problem Claim 15 overcomes. Accordingly, Applicants respectfully request the Examiner withdraw the rejection of independent Claim 15.

Independent Claim 19

On Page 9 of the Office Action, independent Claim 19 has been rejected as obvious under 35 U.S.C. § 103(a) in light of U.S. Patent No. 5,475,843 to Halviatti et al. ("*Halviatti*") in view of U.S. Patent No. 5,634,098 to Janniro et al. ("*Janniro*") and in further view of U.S. Patent No. 5,600,789 to Parker et al. ("*Parker*"). Although Applicants do not necessarily agree with the current rejection of Claim 19, in order to facilitate prosecution of the present application, Applicants have amended Claim 19 to further distinguish the claimed invention from the cited prior art. For example, Applicants have amended Claim 19 to specify a method that involves interrogating a particular Graphical User Interface (GUI) control on a target device by using a first table that includes "***a data field for optionally storing a control-specific identify flag to be used by said test tool for indicating to said test agent that a specific set of properties are to be used in identifying said control on said target device***," and a second table that includes "***a default identify flag to be used by said test tool for indicating to said test agent that a default set of properties are to be used for identifying controls of said particular class of controls on said target device that do not have a control-specific identify flag associated with them, as defined in said first table***."

The Examiner concedes on Page 10 of the Office Action that *Halviatti* fails to disclose a data field for optionally storing a control-specific identify flag to be used by the test tool for indicating to the test agent which of the properties are to be used in identifying that control on the target device. However, the Examiner asserts that *Parker* discloses such a data field.

Parker describes a system for automated testing of GUI applications. The system consists of a test script, test executive, and a test driver that provides the interface to the GUI

application. (See Col. 4, lines 1-20). The test script is directed towards logical objects instead of GUI-specific references (i.e., identify GUI objects at the GUI superclass level). (See Col. 17, lines 2-4). The test executive's primary responsibility is to convert the GUI-independent references from the test script into GUI-specific references. (See Col. 4, lines 7-10). The test driver then takes the GUI-specific references and serves as the interface to the actual objects on the GUI application. (See Col. 4, lines 13-15). Thus, the test script can be written independent of the GUI types, e.g., Windows, Macintosh, and Motif.

The Examiner asserts on Page 10 of the Office Action that *Parker* discloses a data field for optionally storing a control-specific identify flag to be used in identifying that control on the target device. The Applicants note that the optional data field (i.e., [gui-type]) is part of the tag statement. (See Col. 19, lines 18-22). In turn, the tag statement is a part of the means by which the test script is able to uniquely identify the object to which the test script command is referring. (See Col. 18, line 66 to Col. 19, line 2). Thus, though the gui-type is an optional part of the tag statement, the tag statement itself is ***not optional***. The tag statement is a part of the window declaration that allows the test script to manipulate or query a GUI object. (See Col. 18, line 38 to Col. 19, line 17). In contrast, Claim 19 provides greater flexibility and accuracy in testing because it allows the user to specify whether to use specific properties for identifying an object on a GUI application or whether to use default properties in order to identify the object. This flexibility is not provided in *Parker*.

In addition, *Janniro* fails to disclose or suggest any approach to identifying GUI controls on a target device. This is because *Janniro* is not directed to a test tool for testing output rendered to a GUI, but rather is directed to a general software test environment, such as might be used for testing traditional FORTRAN or COBOL based applications. (See Abstract).

Thus, Applicants respectfully assert that *Halviatti, Janniro,* and *Parker* fails to teach or suggest a method that involves identifying a particular Graphical User Interface (GUI) control on a target device by using a first table that includes "***a data field for optionally storing a control-specific identify flag to be used by said test tool for indicating to said test agent that a specific set of properties are to be used in identifying said control on said target device***," and a second table that includes "***a default identify flag to be used by said test tool for indicating to said test***

agent that a default set of properties are to be used for identifying controls of said particular class of controls on said target device that do not have a control-specific identify flag associated with them, as defined in said first table," as recited by independent Claim 19. Accordingly, Applicants respectfully request the Examiner withdraw the rejection of independent Claim 19.

Dependent Claims 13-14 and 20-23

Claims 13-14 depend from independent Claim 9 and therefore include all the limitations of Claim 9 plus additional limitations that further define the invention over the prior art. Claims 20-23 depend from independent Claim 19 and therefore include all the limitations of Claim 19 plus additional limitations that further define the invention over the prior art. Accordingly, for at least the reasons set forth above in regard to independent Claims 9 and 19, Applicant respectfully asserts that these claims are also in condition for allowance.

Conclusion

The foregoing is submitted as a full and complete response to the final Office Action of July 25, 2007. The foregoing amendments to the claims, when taken in conjunction with the appended remarks, are believed to have placed the present application in condition for allowance, and such action is respectfully requested. The Examiner is encouraged to contact Applicant's undersigned attorney at (404) 881-7640 or chris.haggerty@alston.com to resolve any remaining issues in order to expedite examination of the present application.

It is not believed that extensions of time or fees for net addition of claims are required, beyond those that may otherwise be provided for in documents accompanying this paper. However, in the event that additional extensions of time are necessary to allow consideration of this paper, such extensions are hereby petitioned under 37 CFR § 1.136(a), and any fee required therefore (including fees for net addition of claims) is hereby authorized to be charged to Deposit Account No. 16-0605.

Respectfully submitted,

/Christopher S. Haggerty/

Christopher S. Haggerty
Registration No. 58,100

Customer No. 00826
ALSTON & BIRD LLP
Bank of America Plaza
101 South Tryon Street, Suite 4000
Charlotte, NC 28280-4000
Tel Atlanta Office (404) 881-7000
Fax Atlanta Office (404) 881-7777

ELECTRONICALLY FILED USING THE EFS-WEB ELECTRONIC FILING SYSTEM OF THE UNITED STATES PATENT & TRADEMARK OFFICE ON October 25, 2007.